PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY A

Phil. Trans. R. Soc. A (2008) **366**, 3769–3780
doi:10.1098/rsta.2008.0142
Published online 31 July 2008

Security policies and trust in ubiquitous computing

By Anupam Joshi[1,*], Tim Finin[1], Lalana Kagal[2], Jim Parker[1] and Anand Patwardhan[3]

[1]*Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County 1000 Hilltop Circle, Baltimore, MD 21250, USA*
[2]*MIT Computer Science and Artificial Intelligence Lab, Cambridge, MA 02139, USA*
[3]*Cougaar Software, Inc., 7600 Leesburg Pike Ste 105, Falls Church, VA 22043, USA*

Ubiquitous environments comprise resource-constrained mobile and wearable devices and computational elements embedded in everyday artefacts. These are connected to each other using both infrastructure-based as well as short-range ad hoc networks. Limited Internet connectivity limits the use of conventional security mechanisms such as public key infrastructures and other forms of server-centric authentication. Under these circumstances, peer-to-peer interactions are well suited for not just information interchange, but also managing security and privacy. However, practical solutions for protecting mobile devices, preserving privacy, evaluating trust and determining the reliability and accuracy of peer-provided data in such interactions are still in their infancy. Our research is directed towards providing stronger assurances of the reliability and trustworthiness of information and services, and the use of declarative policy-driven approaches to handle the open and dynamic nature of such systems. This paper provides an overview of some of the challenges and issues, and points out directions for progress.

Keywords: ubiquitous computing; security; policy; trust; privacy

1. Introduction and background

Mobile devices with small form factors, yet with computing power comparable to desktops only a few years old, are now common. Their usability has been significantly enhanced by multimodal user interfaces such as touch screens, biometric security devices and accelerometers. The integration of GPS receivers, cameras and recorders in mobile (cell) phones and personal digital assistants has ushered in a new generation of *converged* mobile devices. Computing power is also increasingly embedded in everyday devices, from home appliances to wearable computers. We are now witnessing a proliferation of wireless appliances

* Author for correspondence (joshi@cs.umbc.edu).

One contribution of 19 to a Discussion Meeting Issue ‘From computers to ubiquitous computing, by 2020’.

in everyday life—crib monitors, home security alarms, fire alarm annunciators, surveillance cameras and automobiles. These technological advances are helping to create resource-rich environments in which personal mobile devices can seamlessly integrate—use and provide services. Moreover these mobile devices will be capable of sharing their capabilities via wireless means. Peer-to-peer relationships will enable devices to dynamically form collaborative relationships and perform complex tasks leveraging available resources either shared among the peers or those present in their surrounding infrastructure.

These developments are beginning to introduce new models of distributed communication and computation. Ubiquitous computing demands systems that are *open* in that they do not pre-identify a set of known participants, and *dynamic* in that the participants change regularly, and not just due to occasional failures. It is interesting to note that this evolution is occurring at several levels—communication, infrastructure and application. At the communication level, for example, mobile ad hoc networking protocols such as AODV (Perkins & Royer 1999) treat nodes as autonomous routers, requiring new techniques to protect against malicious or faulty nodes that subvert or black hole packets. Similarly, as applications become more sophisticated and intelligent, they require greater degrees of decision making and autonomy in order to engage and exploit nearby information and services as they move. The long-range vision is described as societies of intelligent, autonomous agents that are goal directed and adaptive; such systems will undergird the ubiquitous computing systems of the future. Even today, we find the new levels of autonomy emerging in infrastructures such as web services and pervasive computing. These systems must exchange information about services offered and sought and their associated security and privacy policies, negotiate for information sharing, and monitor for and report on suspicious or anomalous behaviour.

A new grand challenge thus emerges: securing these open dynamic environments. As a concrete instance, consider providing a secure and privacy enhancing pervasive computing environment in spaces such as an office, hospital, school or subway stop. The space will be filled with devices and agents offering and seeking services. As people move, agents on their personal devices detect, and are detected by, the pervasive infrastructure. The new devices must discover the services of interest to their user from the infrastructure and other devices in the vicinity, negotiate for access, control information exchange, and monitor for suspicious events and report them to the community. Shared knowledge models (ontologies) and norms of behaviour (policies) will undergird the society of cooperating applications, agents and devices. The underlying networking systems will also need to be aware of these norms and the related security and privacy concerns. Addressing this grand challenge will require contributions not just from diverse areas within computer science, but also from other disciplines such as policy, law and various social sciences.

Without appropriate security and privacy mechanisms, these exciting new ideas will be hobbled and the applications they enable will not be deployed or be found socially acceptable. DARPA's LifeLog (Shachtman 2003) program serves as a good example. It was forced to eliminate many of the more exciting possibilities from its scope because adequate privacy mechanisms were not available. We must develop new models for security and privacy that work in such highly distributed, open and dynamic systems. We identify two topics where new

challenges are emerging: computational policies and trust-based security. This paper focuses on these elements, and is not meant as a survey of the entire body of related work in the broad area of security for ubiquitous computing.

(*a*) *Computational policies*

By policy we mean an explicit, executable representation of constraints and rules that govern or inform an agent's or system's behaviour. Policies can define permissions, obligations, norms and preferences for an agent's actions and interactions with other agents and programs. They can factor in not just the parties in the interaction, but even the context factors (such as location, system resources, trustworthiness, use of the information to be shared, etc.) when deciding on the permitted and prohibited actions. Explicit policies, especially those expressed in high-level declarative languages, can be used as the basis for electronic contracts and provide a sublanguage useful for the negotiation of agreements and commitments.

Using explicit policies to manage security, trust and privacy is a promising research area, but one with a number of open challenges. Can we develop meaningful machine interpretable policies for security, digital rights management and privacy that work in open, distributed environments that characterize ubiquitous systems? Can we design policy languages that are simultaneously expressive enough to serve their many needs, intuitive and understandable by humans, and writable by non-programmers? Can we implement policy languages over which we can reason at a high level, answering hypothetical questions about the limitations and vulnerabilities in the security and privacy systems they model (will this policy allow a particular action, and if so under what circumstances)? Can we do all this tractably?

(*b*) *Trust-based security*

Security and privacy based on authentication is not enough in open systems where principals may be able to provide authentication, but are otherwise unknown to the system and hence not authorizable for specific actions. Traditional role-based approaches also fare poorly. Such environments are common on the web and in envisioned pervasive computing environments. A solution is to make security and privacy decisions based on attributes related to trust for which a principal can provide evidence. Human societies use trust and reputation to make decisions about requests for 'service' where a right to that service is not pre-established, and social networks are an important way of transferring trust and reputation. Such societies have overlapping systems of behavioural norms, constraints and rules. We are over-constrained, so we cannot always satisfy all of them, but deviating too much or too often has its consequences—loss of reputation, penalty clauses, imposition of sanctions, etc. These mechanisms need to be understood and computational analogues developed in order for computational agents to better support information sharing and control in human societies.

Creating such mechanisms will involve answering questions such as: can the very human notions of reputation and trust be used by computer applications and agents? How can reputation be managed in scenarios where a centralized authority is not available? Is reputation inherently distributed and emergent

in ubiquitous computing? If so, how can it be managed, maintained, and propagated? How can we build scalable systems that combine traditional authentication-based security regimes with security and privacy decisions based on trust and reputation?

Sharing knowledge is a common problem that both these research challenges need addressed. As our distributed information systems become more ubiquitous, autonomous and complex, there is a stronger need for grounding them on common models of data and knowledge. The agents in such systems need to be able to exchange information, queries and requests with some assurance that they share a common meaning. The lack of a common understanding of shared information opens up new security and privacy vulnerabilities. Monitoring and enforcing security in a distributed system, e.g. for intrusion detection, requires a common model for sharing information about individuals, events and situations. We need better languages in which to define and publish ontologies for security and privacy to support information sharing and cooperation in distributed systems.

Thus far, wireless networking has primarily served to extend the reach of the Internet. Most of the prevalent wireless technologies and their applications are infrastructure based, such as WiFi. In traditional mobile computing environments, devices mostly adhere to the basic client–server model in which the devices act as clients and access non-transient information on trusted servers. In the client–server model, the server is anchored and a client can verify through several authentication and integrity schemes that the information originated from the server, forcing accountability. Mobile devices in ubiquitous computing environments, however, cannot always use such mechanisms to decide the reliability of either the source, or the service or information provided by that source. This is because, in ubiquitous computing, devices interact with others in new environments where they do not have pre-established sources of trust or in a peer-to-peer manner over *mobile ad hoc networks* (MANETs) where a central authority is not accessible. Consequently, devices need a mechanism to evaluate the integrity of their peers and the accuracy of information provided by their peers, as there is otherwise no scheme for protecting a device from malicious peers that deliberately provide unreliable information.

A MANET is a self-organized collection of wireless mobile nodes lacking a fixed network infrastructure and having no central authority. The flexibility and openness of MANETs make them very appealing as an information gathering and exchange medium; however, these two properties can also lead to security vulnerabilities. To fully realize the potential of the mobile ad hoc paradigm, there must be an autonomous approach to mitigating risk and/or place users in control of risk evaluation and usage. Along with enabling devices to estimate their trust in other devices and the accuracy in the information obtained from them, a mechanism must be provided that enables devices to detect and distinguish between *malicious peers*, that purposely provide incorrect information, *ignorant peers*, that are unable to provide reliable information, and *uncooperative peers*, that have reliable information but refuse to make it available to other devices. This mechanism would also implicitly support an *incentive model*, in which all devices must provide only reliable information and provide this information often, otherwise they risk losing the ability to communicate with other devices in the environment.

In MANETs, a server-centric mechanism of identification and authentication is not suitable. Even with limited Internet connectivity, total reliance on conventional security mechanisms involving key distribution centres, certificate authorities or similar forms of remote *trusted* sources imposes serious limitations on the functioning of these devices, in effect limiting them to function only when those remote sources can be contacted. In ubiquitous environments, the number of devices either embedded in the surrounding infrastructure, or personal mobile devices, will be immense. Thus, it is not possible to pre-enumerate all possible devices that may be encountered, nor will it be feasible to centrally register all such devices and then later identify and authenticate them on every encounter.

(i) *Healthcare scenario*

To better illustrate the challenges and issues that face ubiquitous computing, consider a scenario where emergency medical technicians (EMTs) reach the site of a natural disaster to start treatment and triage. The recent devastation caused by Hurricane Katrina comes to mind as an example. The electronic healthcare records (EHRs) of the persons the EMTs treat would typically be stored at access controlled servers, for instance in a hospital, in their doctor's office or potentially in a wearable electronic version of 'medical id bracelets'. Clearly, the traditional access control mechanisms do not help here. It is unlikely that this particular subject (EMT)/object (EHR) combination is pre-identified as requiring an access relationship. Role-based access control (RBAC; Sandhu 1998)-type mechanisms are a popular alternative because RBAC is a NIST standard. However, not every EMT working in the disaster area should get access to the medical records of everyone who is a resident of the area and hence a potential victim, as would typically happen in a pure RBAC case. One could envision a policy which says that EMTs treating a particular person should get access to that person's record while they are actively treating them. We see here the first instance of context and policy modulating access. The context here is a combination of the roles of the individuals and their spatio-temporal relationship.

Of course, the role assignment part of the context assumes trust implicitly—the EMTs would have certificates signed by the local authorities, for which presumably the data sources could verify the signatures. EMTs responding from remote jurisdictions could be trusted because their signing authorities are recognized (a verification chain exists), or because trusted local EMTs vouch for them via (electronic) speech acts such as temporary delegations (Kagal *et al.* 2003*a*,*b*).

The individual, in consultation with her healthcare provider, might have set access control restrictions on her medical records as well (Choudhri *et al.* 2003). She might want EMTs to only look at elements of her EHR that might be needed in an emergency response situation—medications she currently takes, her known allergies, diseases she currently suffers from, doctors who have treated her, etc. These are elements that need to be shared, but only with those involved in her treatment, and only if the situation demands. So, for instance, if the EMT indicates that they need to give her a non-steroidal anti-inflammatory drug (NSAID), the use of sharing the fact that she tolerates naproxen better than diclofenac will be high. On the other hand, if the plan is to administer Tylenol,

then this information need not be shared. She might not want an EMT to know her cholesterol levels, or the fact that she had suffered from clinical depression in the past, that necessitated counselling. Such need to know principals can also be captured by a policy.

The hospital that has some of her records might similarly specify how they can be shared. It might be willing to share information about what treatment she was given, or what surgeries were performed on her, but might not want to reveal who was the attending physician (*note that this conflicts with the patient's policy*), or whether there were any medical errors and so on except to physicians who are employed by the hospital or have admitting privileges there. It may want to further restrict some elements of the record to be only accessible from machines connected to the hospital's secure network. So, we now have the context expanded to involve roles, spatio-temporal relationships, trust, access devices, privacy and multiple potentially inconsistent policies.

Some patients are released after preliminary treatment, but others may have serious trauma injuries that require emergency evacuation and surgical intervention. As the patient is medivaced, the hospital record system might decide that more information now needs to be shared—past surgical history, all known allergies, all current medications, prescription or otherwise, past experiences under anaesthesia, etc. The idea is to make this available to the critical care team, i.e. planning the surgery. This information may be more than the policies of the hospital and/or the patient will normally share. The record system might then need to reason over the consequences of violating the sharing policy (revealing sensitive information) versus its use (saving a patient's life). It might decide to release the requested information, but only after the agent representing the critical care team makes a(n electronic) speech act undertaking the obligation not to share it further and securely erase it after it is no longer needed. Alternatively, using its knowledge base of medical information, it might infer the least further information over and above that is allowed by policy and needed to be shared in order to perform the task at hand.

2. Computational policies

Security and privacy research for distributed systems have mostly focused on models in which clients, services and mediators, while physically distributed and potentially subject to disconnections, nevertheless assume that the entities are predetermined, relatively static and share the same domain knowledge and security and privacy infrastructure. This is clearly not an assumption that is realistic for ubiquitous computing. We propose that a declarative policy-based approach to be used, where the norms or rules of ideal behaviour of entities in these environments are described in a machine-understandable specification language. These policies describe what an entity can or must do in a certain context and allow the behaviour of entities to be modified without affecting the underlying mechanisms and architecture. Along with providing the openness required in these environments, this approach also provides greater autonomy as entities can choose to accept or deny a particular norm. Policy-driven systems can even be engineered so as to be extremely lightweight on resource-constrained devices (Patwardhan *et al.* 2004).

By policy, we mean an explicit representation of constraints and rules that govern an agent's or system's behaviour. As an example, EMTs should not call for air evacuation unless a patient will be fatally harmed, if they cannot be operated upon in the next hour. This policy causes the EMTs to behave differently with different patients. Policies can define what is permitted or prohibited, what is preferred of the permitted actions, and what obligations must be incurred in order to undertake an action. Such policies, when expressed in languages that the other party in an interaction can understand, can be used to negotiate permissions where none existed *a priori*.

Rei (Kagal *et al.* 2003*b*), a policy language developed by our group, is an example of a *declarative policy language* that uses Semantic Web technologies to describe policies as constraints over allowable and obligated actions on resources in the environment. Rei allows policies to be described over heterogeneous domain information defined in one of the Semantic Web languages providing common understanding between participants who may not use the same information model. Rei is suitable for ubiquitous computing, because it allows policies to be described in terms of attributes of users, actions and other contexts instead of identities, and it provides greater extensibility as policies can be described over domain knowledge at different levels of abstraction (Kagal *et al.* 2006). It also supports speech acts that allow for dynamic modification of rights by those authorized to alter them—for instance, by delegating rights.

Rei also supports sanctions and conditional permissions that are common in human societies. Users are usually over-constrained and cannot always satisfy all of the policies. There are consequences associated with deviation from a policy. Rei allows these consequences to be modelled as *sanctions* so that autonomous entities or providers can reason over them to decide whether or not to deviate from a certain policy. Consider an example. The storage system for medical records must not disclose to the EMT that the patient they are treating has a disease such as HIV, if the patient is being treated for an unrelated ailment. If it does disclose this information, it will no longer be deemed HIPAA compliant. Conditional permissions allow an entity to perform a certain action or a set of actions under the condition that it will take on certain additional responsibilities. These conditional permissions impose additional obligations on the entity after the permission is exercised. For example, if an EMT refers a patient for secondary care, they must delegate to the care provider the right to access their notes about the patient.

Rei is based on deontic concepts including permissions, obligations, claims, prohibitions and dispensations. These policies will describe what the ideal behaviour for an entity is in a certain context. For example, the constraint 'You must use an encrypted channel when transferring medical data' can be modelled as an appropriate behaviour for an entity (agent, web service and human user) that wants to use a service. The entity is 'permitted' to access the service if it meets a certain condition, i.e. uses encrypted communication. It can be described as an access control policy for the service. However, these policy specifications should not only be able to represent security, but all aspects of behaviour including privacy, management, conversation, etc. Another example is 'You may request location data from the GPS device'. This can represented as a 'privilege' or a 'claim', but it again represents the ideal behaviour of the entity. Negative modalities should also be possible. For example, 'You should not use my location

details for pushing services or products' prohibits the entity (in this case, a location-based directory service) from performing a certain action. 'Authentication is not required if the connection is from within the firewall' is a dispensation that frees the entity from the obligation of authenticating itself.

Rei is, of course, just one of the recent efforts to develop declarative policy languages. Most are not, unlike Rei, motivated by the security and privacy issues in open systems, such as ubiquitous computing. These include industry standards such as XACML (Moses *et al.* 2005), but also academic efforts ranging from more practical implemented languages such as Ponder (Damianou *et al.* 2001) to theoretical languages (Jajodia *et al.* 1997) and finally to other Semantic Web-based languages such as KAoS (Tonti *et al.* 2003). We argue that policy languages grounded in Semantic Web technologies such as RDF and OWL allow policies to be described over heterogeneous domain data and promote common understanding among the participants who might not use the same information model. A common problem with many of these languages is that they tend not to be defined in terms of a model recognized by the security community that impedes their adoption. Researchers in the ubiquitous computing community need to understand what security models are best suited to our domain—recent developments on usage control models (Sandhu & Park 2003) sound promising. Policy languages then need to be developed that can be grounded on such models. Very recently, we have shown how the well-known RBAC security model can be captured using an ontology and a set of rules in OWL (Finin *et al.* 2008).

3. Trust-based security

Authentication-based security and privacy is not enough in open systems that characterize ubiquitous computing. The fact that an entity can identify itself is clearly of no use, when it is in an environment that is not its home. In our example, the rights of visiting EMTs cannot be solely established based on their foreign credentials. Traditional role based approaches also fare poorly, as our example illustrated. A possible approach is to make *security and privacy decisions based on attributes related to trust* for which a principal can provide evidence. These include self-evident properties (ones that any observer can reliably sense such as a request originated from a .mil host—something established using SSL and a trusted DNS), proof of key attributes, signed statements from a trusted source delegating a permission, or undertaking an enforceable obligation in return for access (agreeing to forward packets for the next 10 min). There is a large body of work exploring trust issues in traditional distributed systems. For some representative examples, see Blaze *et al.* (1996) and Grandison & Sloman (2000).

Social networks are an important way of transferring trust and reputation in human societies (Golbeck *et al.* 2003). We propose the concept of *pack formation* that uses accounts of prior encounters, evolved trust and recommendations, to form local packs. These packs are analogues of the human social networks that can help to transfer trust and reputation. As in real human societies, the societies of entities that participate in a ubiquitous computing environment have overlapping systems of behavioural norms, constraints and rules. We are over-constrained, so we cannot always satisfy all of them, but deviating too much or

too often has its consequences—loss of reputation, penalty clauses, imposition of sanctions, etc. These mechanisms need to be understood, and computational analogues for reputation management and sanctions need to be developed, in order for computational agents to better support information sharing and control in human societies.

The networking layers, especially MANETs that form an important component of ubiquitous computing, can benefit from knowing who the reliable or trusted peers are within the local neighbourhood for preferential consideration in forming routes and for peer discovery (Buchegger & Boudec 2002). The application layer can benefit from reports of malicious activity detected by the lower layers and appropriately modify their trust assessments. Further recommendations by trusted devices can be then used to create new trust relationships or modify existing ones. Moreover, the networking layer can be useful in the sanction process as well. For instance, nodes can refuse to route packets for others that are deemed not to be trustworthy. This could be for a variety of reasons, such as because they are intruders, or not following policies.

Connectivity provided by ad hoc networking entails that the peers in the pervasive environment are cooperative. While this is well understood for the network layer (each node acts as a router), it holds true for other layers as well. For instance, consider securing such networks. Owing to the security threats posed to individual mobile devices, collaborative efforts in countering intrusive behaviour are required. Since the scope of intrusion detection mechanisms deployed on individual devices is limited to their radio-range, collaborative mechanisms are required for communicating suspicious activity and intrusions to other devices in the vicinity. Complex processes of trust evolution can also be simplified using recommendations from trusted peers—another motivating factor for forming local collaborative groups such as *Packs*.

In a closed environment where identities can be tightly bound to entities, one could envision trust and reputation issues being centrally managed. However, in ubiquitous computing environments, it is of necessity that devices will interact with other devices or environments that are 'foreign', for which even the identities proffered when requesting access are potentially suspect. In such situations, not only would we need policies that locally define interaction norms, but also reputations that each entity might manage locally.

We propose giving MANET nodes the ability to independently evaluate *trust* in the nodes with which they interact. This solution involves a reputation management system through which nodes can evaluate, maintain and distribute information about trust relationships within a MANET. Each node can make autonomous decisions about the trustworthiness of other nodes, providing an alternative to third-party authentication during periods of disconnection. Unlike wired networks, each node within a MANET has the ability to handle data through routing protocols. That being the case, there is a potential for nodes to act in a malicious manner. Commonly, the communication links of MANET nodes are symmetric with respect to transmit and receive range. If node A is within radio range of node B, then it is expected that each can receive the other's transmissions. Furthermore, if node A transmits a packet to node B (not being the final destination), then node A can promiscuously snoop and listen for node B to forward the packet. This simple concept has led to some very interesting research where promiscuous snooping has the ability to provide the 'eyes and

ears' to application level trust agents. Not only nodes along a transmission route, but neighbouring nodes within the range of that route can participate in the snooping (Marti *et al.* 2000; Parker *et al.* 2004). Packet fields can be monitored and then matched to snooped output packets from a node. Such tracking can detect packet modification, dropping or misrouting. Thresholds determine the boundary between acceptable and unacceptable behaviours.

Determining the thresholds comes in large part through the context within which the MANET exists. Slow-moving nodes within MANETs may have long connection times resulting in stable network routes. Fast-moving nodes bring the problem of frequent disconnection and overhead to constantly update broken routes during data flow. In general, the more traffic there is within a MANET, the more difficult it becomes for nodes to handle traffic and act as malicious detection entities.

The biggest problem in these environments is false detection. Mobility can cause the detection of packet mishandling, when, in fact, a node may have simply moved out of range. Likewise, RF collisions at the monitoring node in a noisy network may cause false detection, when, in fact, the packet was successfully and correctly forwarded. We believe that false positives can be mitigated using cross-layer analysis (Parker *et al.* 2006). Packet modifications at the transport layer, rerouting of packets at the network layer, and CTS/RTS attacks to dominate bandwidth usage at the MAC layer are all examples of malicious activity. If any one of these attacks is used, then standard thresholds can be used to detect the malicious activity. However, it is possible for a malicious node to selectively disrupt transmissions by using a combination of these techniques while staying below threshold values at each layer. If threshold values are dropped, the false detection rate increases. However, detection data from across these layers can be aggregated together to reveal malicious activity from a single node or combination of nodes without lowering the threshold at any individual layer.

Ding *et al.* (2003) propose using two kinds of trust, *viz.* 'domain trust' and 'referral trust.' *Packs* are also useful here. Trusted peers in a pack can provide reputations for nodes they may have previously encountered to others in their pack that have not encountered them. Nodes can also ask other trustworthy nodes for providing information (domain trust) or trust them to provide referrals to other devices that might have that information (referral trust). These can be used, in conjunction with an entity's own observations at various layers of connectivity, and its local policies, to determine the trustworthiness of another entity in the ubiquitous computing environment (Perich *et al.* 2004; Patwardhan *et al.* 2006).

Consider the privacy policy of a patient in our example, who is being treated by EMTs. 'EMTs should let doctors at hospitals that are affiliated with the Louisiana State University access all my records. They may also share my records with other doctors needed during the emergency as long those doctors undertake to share their records about me with my primary physician.' The first part of the policy requires a doctor to prove that they are part of a hospital (signed statement from hospital, certificate from hospital board, or a delegation from the hospital administrator) and then provide information that proves that the hospital is on a particular list. The second part of the policy is a conditional policy relating to the same doctor. It provides the additional permission to share the information as long as the associated obligation is met in the future.

However, to decide if another entity is trustworthy enough to take its assurance about an obligation, the EMT might use observations about it, or ask other trusted nodes in the neighbourhood who may have interacted with the unknown entity in the past.

4. Conclusion

Ubiquitous computing is best viewed as an open, distributed system where not all principals are pre-identified, and the cohort that an entity interacts with dynamic changes. As we argue in this paper, many existing approaches to security and privacy do not work well in such environments. We further suggest that declarative policies described in Semantic Web languages can be combined with decentralized trust and reputation management systems and network-level observations to address the challenges in security and privacy that ubiquitous computing systems pose. To support this contention, we describe the recent works in this area by our group and others.

This work was supported in part by NSF awards 0716627, 0325172 and 0242403, and support from IBM.

References

Blaze, M., Feigenbaum, J. & Lacy, J. 1996 Decentralized trust management. In *IEEE Conf. on Privacy and Security*.

Buchegger, S. & Boudec, J. L. 2002 Performance analysis of the confidant protocol: cooperation of nodes: fairness in distributed *ad hoc* networks. In *Proc. IEEE/ACM Workshop on Mobile Ad Hoc Networking and Computing* (*MobiHOC*).

Choudhri, A., Kagal, L., Joshi, A., Finin, T. & Yesha, Y. 2003 PatientService: electronic patient record redaction and delivery in pervasive environments. In *5th Int. Workshop on Enterprise Networking and Computing in Healthcare Industry* (*Healthcom 2003*).

Damianou, N., Dulay, N., Lupu, E. & Sloman, M. 2001 The ponder policy specification language. In *Proc. Int. Workshop on Policies for Distributed Systems and Networks, LNCS*, pp. 18–37.

Ding, L., Zhou, L. & Finin, T. 2003 Trust based knowledge outsourcing for semantic web agents. In *Proc. 2003 IEEE/WIC Int. Conf. on Web Intelligence*.

Finin, T., Joshi, A., Kagal, L., Niu, J., Sandhu, R., Winsborough, W. & Thuraisingham, B. 2008 Role based access control and OWL. In *Proc. 4th Int. Workshop on OWL: experiences and directions*.

Golbeck, J., Parsia, B. & Hendler, J. 2003 Trust networks on the semantic web. In *Proc. Cooperative Information Agents VII*, vol. 2782. Lecture Notes in Computer Science, pp. 238–249.

Grandison, T. & Sloman, M. 2000 A survey of trust in internet application. *IEEE Commun. Surv. Tutorials* (*Fourth Quarter*) **3**.

Jajodia, S., Samarati, P. & Subrahmanian, V. S. 1997 A logical language for expressing authorizations. In *1997 IEEE Symposium on Security and Privacy* 00:0031.

Kagal, L., Finin, T. & Joshi, A. 2003*a* A policy based approach to security for the semantic web. In *2nd Int. Semantic Web Conference* (*ISWC2003*).

Kagal, L., Finin, T. & Joshi, A. 2003*b* A policy language for a pervasive computing environment. In *Proc. IEEE 4th Int. Workshop on Policies for Distributed Systems and Networks*.

Kagal, L., Finin, T., Joshi, A. & Greenspan, S. 2006 Security and privacy challenges in open and dynamic environments. *Computer* **39**, 89–91. (doi:10.1109/MC.2006.207)

Marti, S., Giuli, T., Lai, K. & Baker, M. 2000 Mitigating routing misbehavior in mobile *ad hoc* networks. In *Proc. ACM MOBICOM 2000*.

Moses, T. *et al.* 2005 *eXtensible Access Control Markup Language* (XACML) Version 2.0. *OASIS Standard* 200502.

Parker, J., Undercoffer, J., Pinkston, J. & Joshi, A. 2004 On intrustion detection and response for mobile *ad hoc* networks. In *Proc. 23rd IEEE International Performance, Computing, and Communications Conference* (*IPCCC 2004*), pp. 747–752. Phoneix, AZ: IEEE Computer Society.

Parker, J., Patwardhan, A. & Joshi, A. 2006 Cross-layer analysis for detecting wireless misbehavior. In *Proc. IEEE Consumer Communications and Networking Conference* (*CCNC 2006*), vol. 1, pp. 6–9. Las Vegas, NV: IEEE Computer Society.

Patwardhan, A., Korolev, V., Kagal, L. & Joshi, A. 2004 Enforcing policies in pervasive environments. In *Int. Conf. on Mobile and Ubiquitous Systems: Networking and Services*. Cambridge, MA: IEEE.

Patwardhan, A., Joshi, A., Finin, T. & Yesha, Y. 2006 A data intensive reputation management scheme for vehicular *ad hoc* networks. In *Proc. 2nd Int. Workshop on Vehicle-to-Vehicle Communications*. Cambridge, MA: IEEE.

Perich, F., Undercoffer, J. L., Kagal, L., Joshi, A., Finin, T. & Yesha, Y. 2004 Reputation we believe: query processing in mobile *ad-hoc* networks. In *Int. Conf. on Mobile and Ubiquitous Systems: Networking and Services*.

Perkins, C. & Royer, E. 1999 *Ad hoc* on-demand distance vector routing. In *IEEE Mobile Computing Systems and Applications*.

Sandhu, R. S. 1998 Role-based access control. In *Advances in computers*, vol. 48 (ed. M. Zerkowitz), pp. 237–286. San Diego, CA: Academic Press.

Sandhu, R. & Park, J. 2003 Usage control: a vision for next generation access control. In *Proc. Computer Network Security*, vol. 2776. Lecture Notes in Computer Science, pp. 17–31.

Shachtman, N. 2003 Pentagon alters Lifelog project. See http://www.wired.com/politics/law/news/2003/07/59607. Wired News.

Tonti, G., Bradshaw, J., Jeffers, R., Montanari, R., Suri, N. & Uszok, A. 2003 Semantic web languages for policy representation and reasoning: a comparison of kaos, rei, and ponder. In *Proc. Int. Semantic Web Conference*, vol. 2870. Lecture Notes in Computer Science, pp. 419–437.